WE CLAIM:

1. A method for computing an optimal route between a first network element (NE) and other NEs of a network using a weighted graph of interconnected nodes that represent the network, the optimal route being subject to a subset sequence constraint associated with a serial restriction group in the weighted graph, the method comprising:

creating a list of temporary labels respectively associated with the nodes of the graph, each node being assigned a primary label, and each node in the serial restriction group being additionally associated with a backup label, the list initially comprising a most optimal primary label of a root node that represents the first NE;

examining the list of temporary labels to identify a most optimal label, and making the identified label permanent to remove it from the list;

selectively updating primary labels in the list for every node adjacent the permanently labeled node in order to ensure that the primary labels identify an optimal allowable path from the root node to the adjacent node; and

repeating the examining and selectively updating until all primary labels of nodes representing the other NEs are permanent.

2. The method as claimed in claim 1 wherein selectively updating further comprises:

selectively updating backup labels in the list for every node adjacent the permanent labeled node in

order to ensure that the backup labels identify an optimal allowable path from the root node to the adjacent node that is allowably extended to another node in the serial restriction group.

3. The method as claimed in claim 2 wherein creating comprises initializing each of the temporary labels to include an optimization parameter that is least optimal, and an identifier of a null path.

4. The method as claimed in claim 3 wherein the initializing each of the temporary labels comprises assigning a cost, which is the optimization parameter, and wherein the examining the list comprises selecting a least cost temporary label.

5. The method as claimed in claim 3 wherein selectively updating the backup labels comprises ensuring that a backup label of the adjacent node is initialized unless the primary label path of the adjacent node cannot be allowably extended to a node in the serial restriction group, in accordance with the subset sequence constraint.

6. The method as claimed in claim 5 wherein selectively updating the backup labels further comprises setting a restriction flag at the adjacent node and saving a previous primary label as the backup label when the primary label is updated to identify a path that is not allowably extended to another node in the serial restriction group, and further comprising unsetting the restriction flag at the adjacent node and reinitializing the backup label of the adjacent node if the primary label of the adjacent node is

subsequently updated to identify a path that is allowably extended to another node in the serial restriction group.

7. The method as claimed in claim 6 wherein selectively updating the primary label further comprises determining whether a path of the permanently labeled node extended to the adjacent node is allowable by the subset sequence constraint using at least one of: a rule for inclusion of members in the subgroup; and a list identifying allowable sequences of nodes.

8. The method as claimed in claim 7 wherein the subset sequence constraint includes subset intransitivity, and the determining whether the path is allowable comprises determining that the path of the permanently labeled node extended to the adjacent node is not allowable if the adjacent node is in the serial restriction group, and a restriction flag is set at the permanently labeled node.

9. The method as claimed in claim 8 wherein the subset sequence constraint includes a subset seriality restriction which precludes paths that include two links between three nodes in the serial restriction group, and wherein the determining whether the path is allowable further comprises determining that a path of permanently labeled node extended to the adjacent node is not allowable if the adjacent node is in the path of permanently labeled node.

10. The method as claimed in claim 7 wherein:

the allowable paths are subjected to a plurality of subset sequence constraints, associated with respective disjoint serial restriction groups;

initializing further comprises indicating each node's membership in a serial restriction group by assigning a one of a plurality restriction flags uniquely associated with respective serial restriction groups; and

determining whether the path is allowable further comprises determining whether any of the subset sequence constraints precludes the inclusion of members in the corresponding serial restriction groups.

11. A method for deriving an optimal route from a first network element (NE) in a network to other NEs in the network, wherein the route is subject to a subset sequence constraint, the method comprising:

obtaining a weighted graph representing the network, the graph comprising nodes representing NEs in the network, a subset of the nodes being identified as members of a serial restriction group; and

constructing a spanning tree over the graph rooted at a root node which represents the first NE by:

iteratively expanding the tree to include a node to which a path from the root node is allowable in accordance with the subset sequence constraint, and is most optimal among the paths to nodes outside of the tree; and

if the path to the node cannot be extended to a node in the serial restriction group, reincluding the node through a secondary path to the node that can be extended to a node in the serial restriction group, when the secondary path is most optimal among the paths to nodes outside of the tree.

12. The method as claimed in claim 11 wherein constructing the spanning tree further comprises:

initializing a set of primary labels, each associated with a respective node;

initializing a set of backup labels, each associated with a respective node in the serial restriction group; and

setting an optimization parameter of the label of the root node to a most optimal value.

13. The method as claimed in claim 12 wherein:

including a node in the tree comprises permanently labeling the node and updating temporary labels of nodes adjacent to the permanently labeled node; and

iteratively expanding further comprises choosing the node that is most optimal from among the temporary labels.

14. The method as claimed in claim 13 wherein:

updating temporary labels comprises:

ensuring that a primary label of the node identifies an optimal allowable path from the root node to the labeled node; and

if the identified path to the node is not allowably extended to a node in the serial restriction group, ensuring that a backup label of the node identifies an optimal allowable path from the root node to the labeled node that is allowably extended to a node in the serial restriction group; and

re-including a node comprises making permanent a backup temporary label.

15. The method as claimed in claim 14 further comprising:

determining whether a path of a permanent label extended to a node is allowable by the subset sequence constraint using at least one of a rule, and a list identifying allowable sequences of nodes;

not updating a temporary label of the node if the path is not allowable; and

setting a restriction flag at the node if the temporary label is a primary label that is changed because the path is allowable, but is not allowably extended to another node in the serial restriction group.

16. The method as claimed in claim 15 wherein the subset sequence constraint includes subset intransitivity, and the determining whether the path is allowable comprises determining that a path to a node **n** is not allowably extended to an adjacent node **m** in the serial restriction group, if the restriction flag is set at **n**.

17. The method as claimed in claim 16 wherein the subset sequence constraint includes a subset seriality restriction which precludes routes that follow two links between three NEs that correspond to nodes in the serial restriction group, and the determining whether the path is allowable further comprises determining that a path to a node **n** extended a node **m** is not allowable if **m** is in the path to **n**.

18. The method as claimed in claim 16 wherein a plurality of disjoint serial restriction groups are defined over nodes in the graph, and initializing further comprises indicating each node's membership in a serial restriction group by defining restriction flags uniquely associated with respective serial restriction groups.

19. A route selection processor adapted to use a weighted graph representing a network of network elements (NEs) to identify an optimal route from a first network element (NE) of the network, to other NEs of the network, subject to a subset sequence constraint, the route selection processor adapted to:

construct a spanning tree over the graph rooted at a root node representing the first NE by:

iteratively expanding the tree to include a node of the graph to which a path from the root node is allowable, in accordance with the subset sequence constraint, and is most optimal among the paths to nodes outside of the tree; and

if the path to the node cannot be extended to a node in a serial restriction group that

includes some of the nodes in the graph, re-including the node through a secondary path to the node that can be extended to a node in the serial restriction group, when the secondary path is most optimal among the paths to nodes outside of the tree.

20. A route selection processor as claimed in claim 19 wherein the subset sequence constraint includes a subset intransitivity limitation on allowable paths, and a determination that a path to a node is not allowable is made by preventing a path from extending over three consecutive nodes in the serial restriction group.

21. A route selection processor as claimed in claim 20 wherein the subset sequence constraint precludes routes that have two links between two of three NEs that are represented by nodes in the serial restriction group, and a determination that a path to a node is not allowably extended to an adjacent node is made by preventing the adjacent node from being labeled if it is included in the path.